Graph Algorithms in Bioinformatics and Genome Assembly

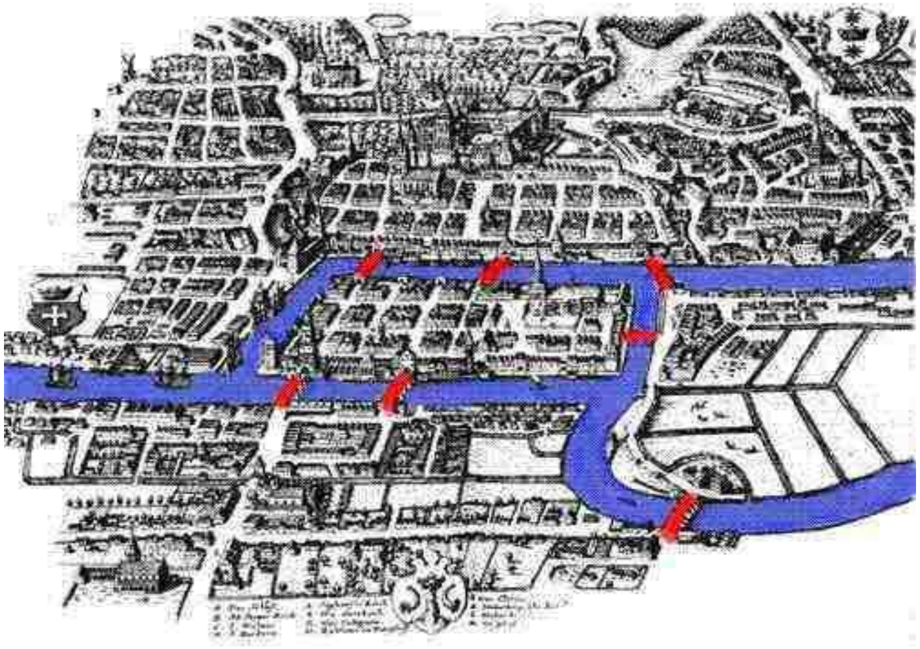
Outline

- Introduction to Graph Theory
- Eulerian & Hamiltonian Cycle Problems
- Benzer Experiment and Interal Graphs
- DNA Sequencing
- The Shortest Superstring & Traveling Salesman Problems
- Sequencing by Hybridization
- Fragment Assembly and Repeats in DNA
- Fragment Assembly Algorithms

The Bridge Obsession Problem

Find a tour crossing every bridge just once

Leonhard Euler, 1735

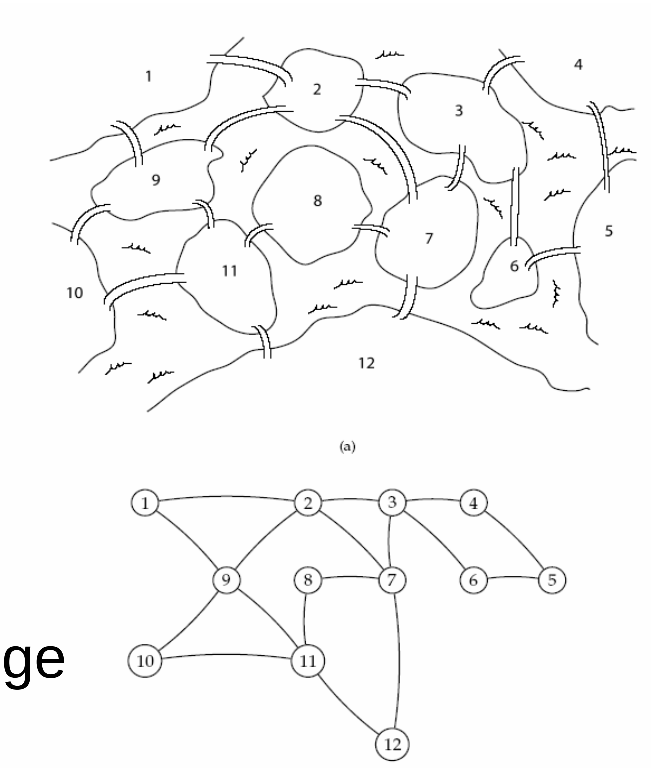


Bridges of Königsberg

Eulerian Cycle Problem

- Find a cycle that visits every ***edge*** exactly once

- Linear time



(a)

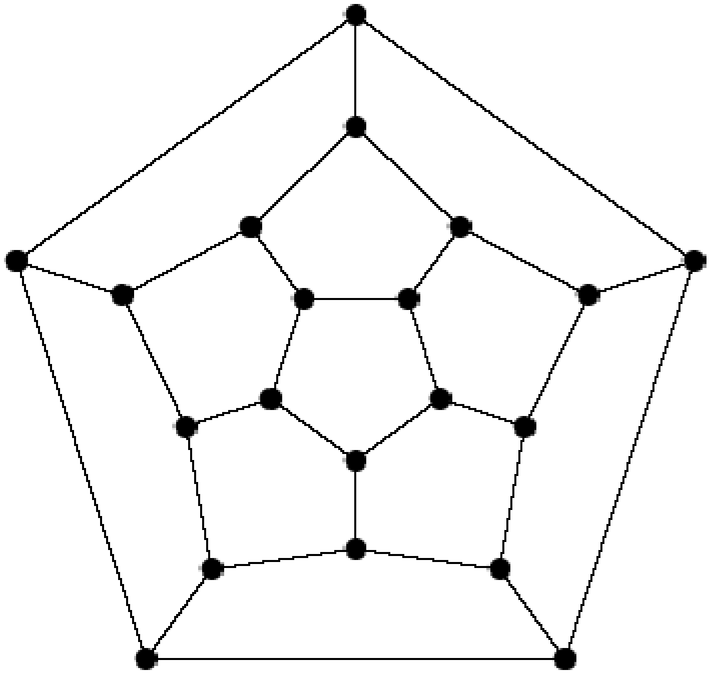
More complicated Königsberg Bridge Walking Problem

Solution: 1 2 3 4 5 6 3 7 2 9 11 8 7 12 11 10 9 1
Note, all vertices have even degree.

Hamiltonian Cycle Problem

- Find a cycle that visits every ***vertex*** exactly once

- NP – complete

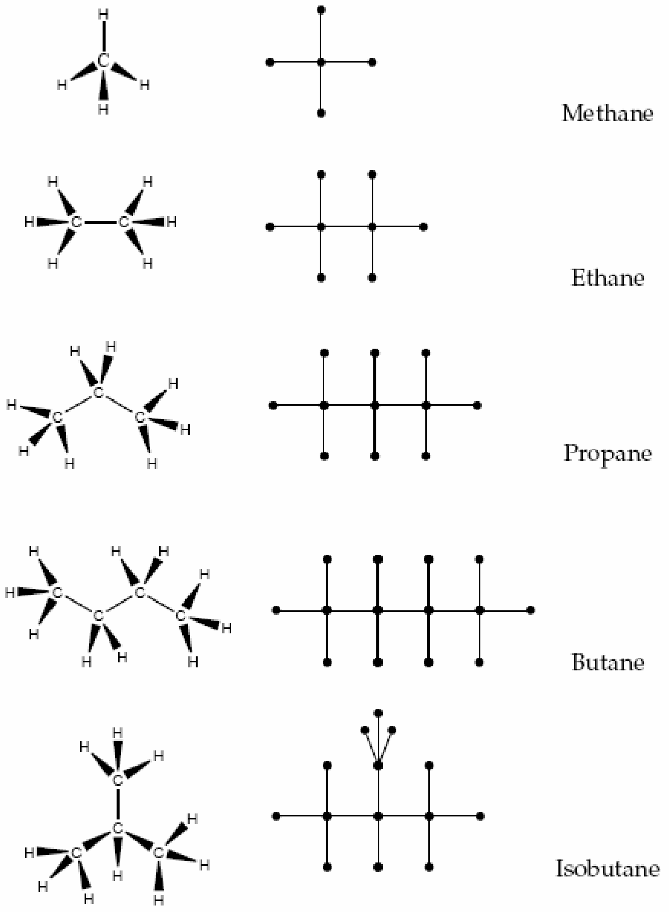


Game invented by Sir William Hamilton in 1857

Mapping Problems to Graphs

Arthur Cayley studied chemical structures of hydrocarbons in the mid-1800s

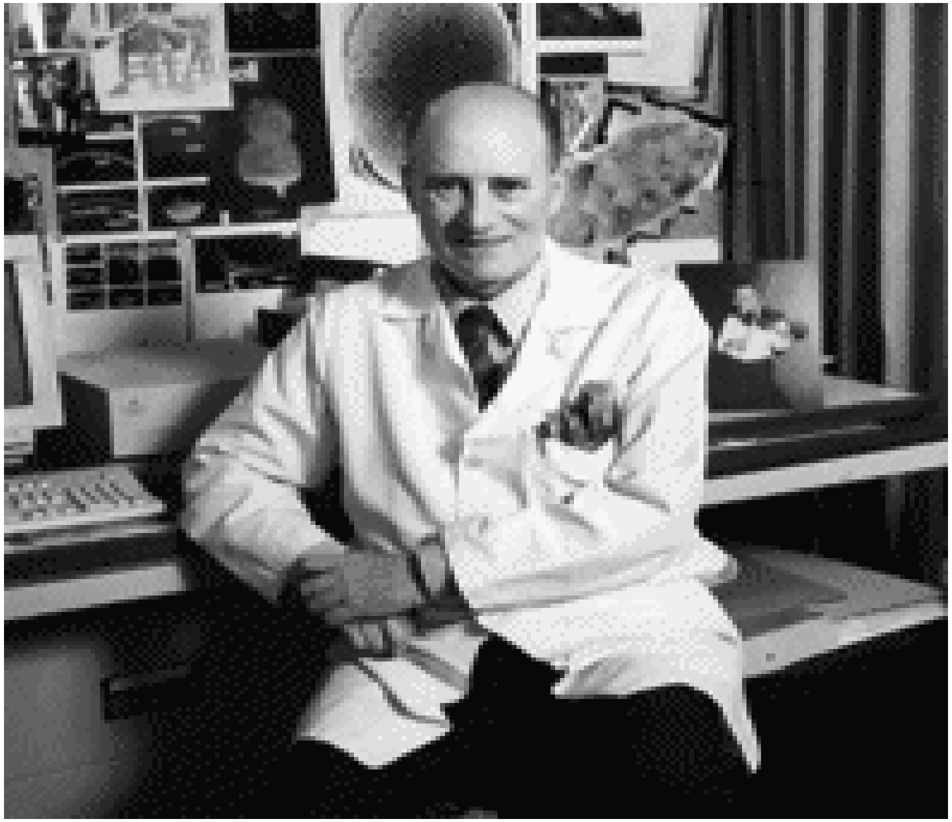
- He used **trees** (acyclic connected graphs) to enumerate structural isomers. A tree with n
- vertices must have n-edges. Caley used this property to find structural isomers.



Beginning of Graph Theory in Biology

Benzer's work

- Developed deletion mapping
- "Proved" linearity of the gene
- Demonstrated internal structure of the gene



Seymour Benzer, 1950s

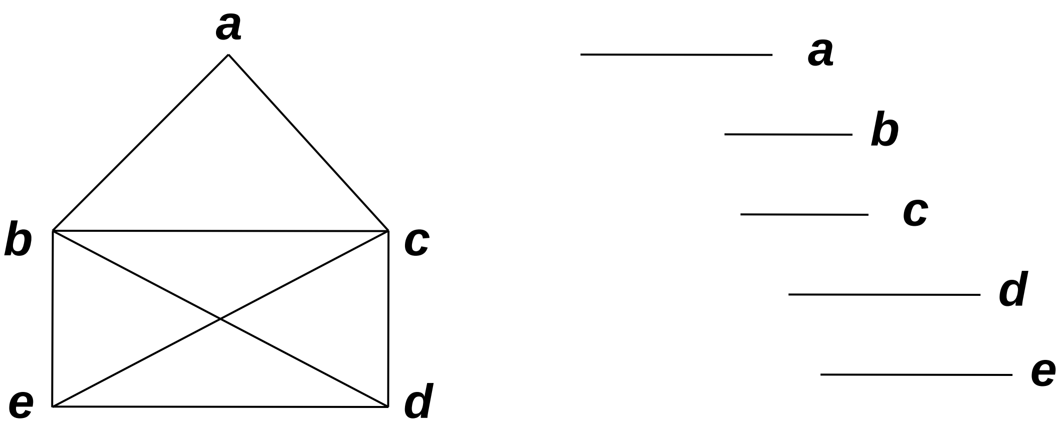
Viruses Attack Bacteria

- Normally bacteriophage T4 kills bacteria
- However if T4 is mutated (e.g., an important gene is deleted) it gets disable and looses an ability to kill bacteria
- Suppose the bacteria is infected with two different mutants each of which is disabled – would the bacteria still survive?
- Amazingly, a pair of disable viruses can kill a bacteria even if each of them is disabled.
- How can it be explained?

Benzer's Experiment

- Idea: infect bacteria with pairs of mutant T4 bacteriophage (virus)
- Each T4 mutant has an unknown interval deleted from its genome
- If the two intervals overlap: T4 pair is missing part of its genome and is disabled – bacteria survive
- If the two intervals do not overlap: T4 pair has its entire genome and is enabled – bacteria die

Interval Graph G=(V,E)

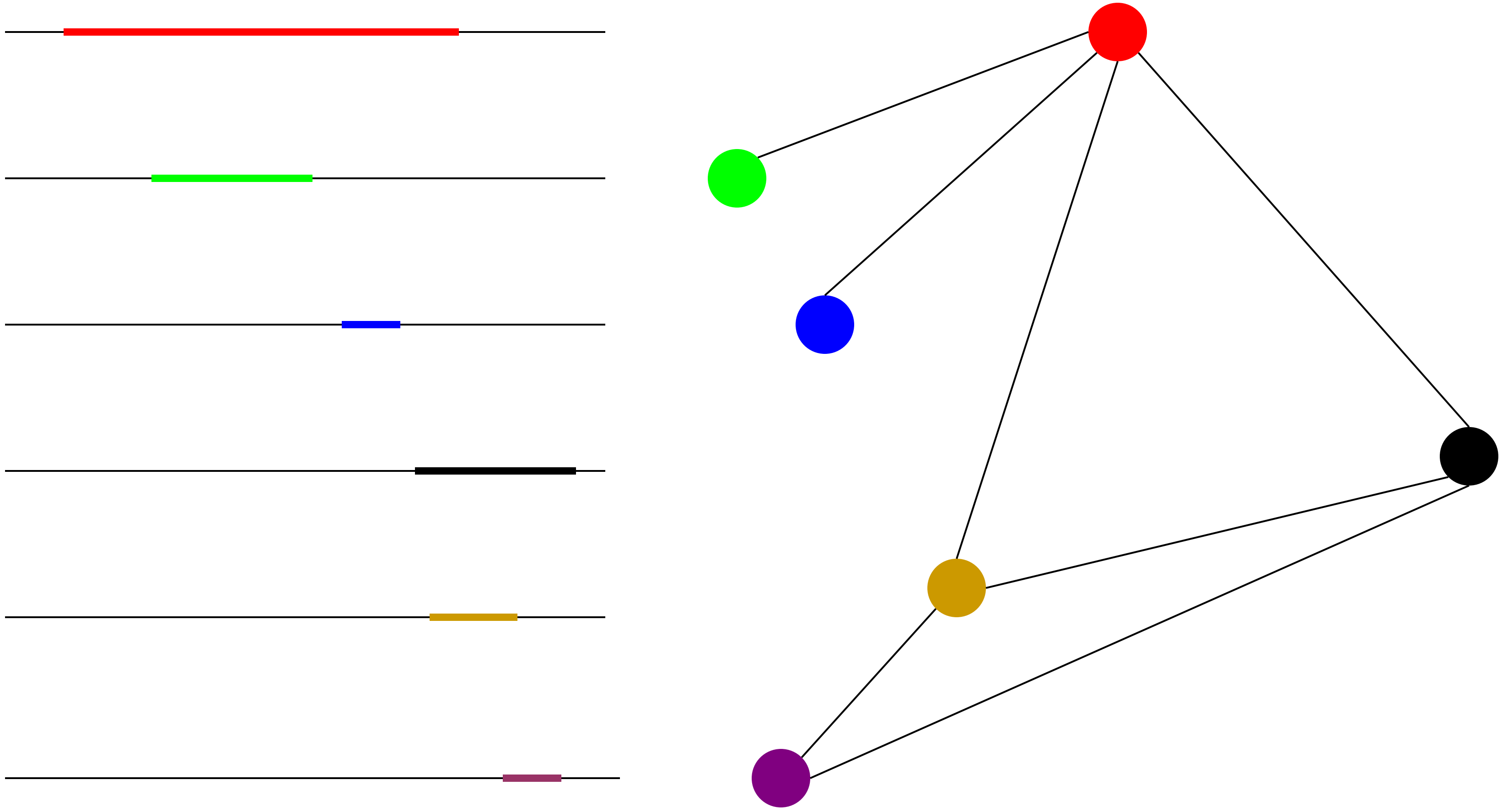


V= set of vertices represent intervals
E= set of edges
There is an edge between two vertices v and w in V
if and only if the intervals v and w overlap.

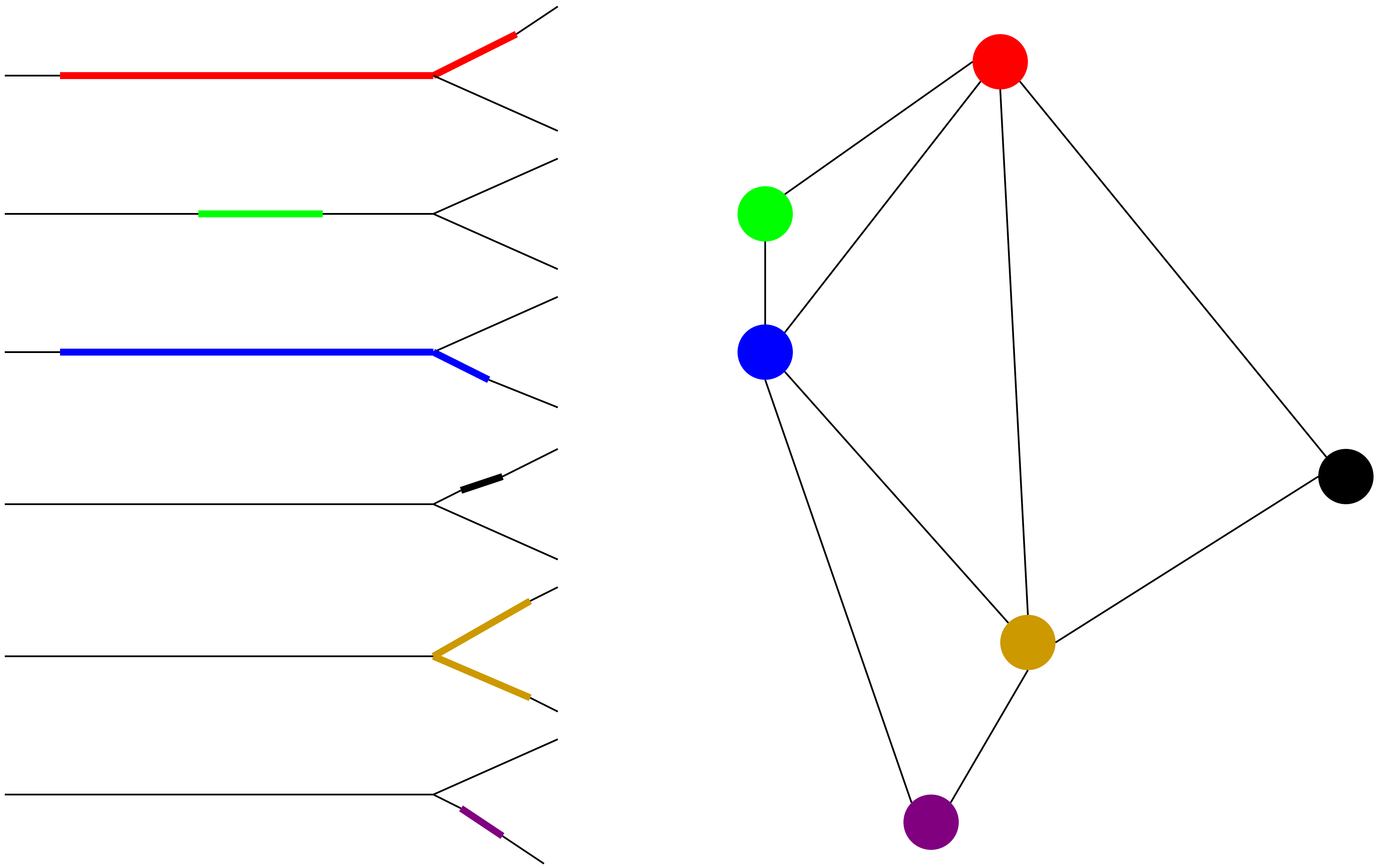
Benzer's Experiment and Graphs

- Interval graph structure reveals whether DNA is linear or branched DNA. Caley reasoned that if genes were linear, he would probably see one kind of graph, but if the genes were branched, he would see a different graph.

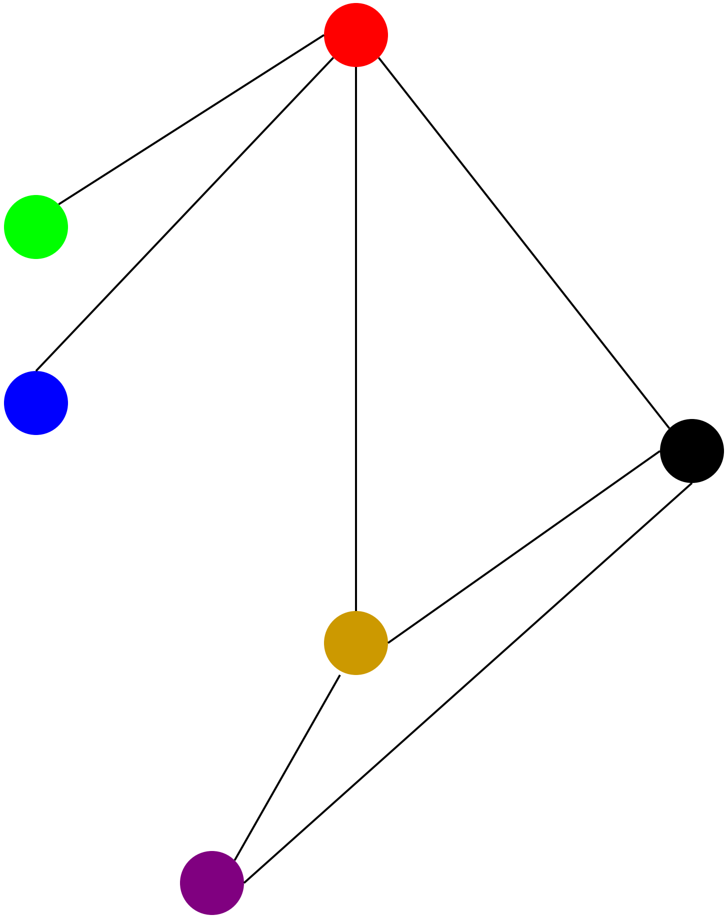
Interval Graph: Linear Genes



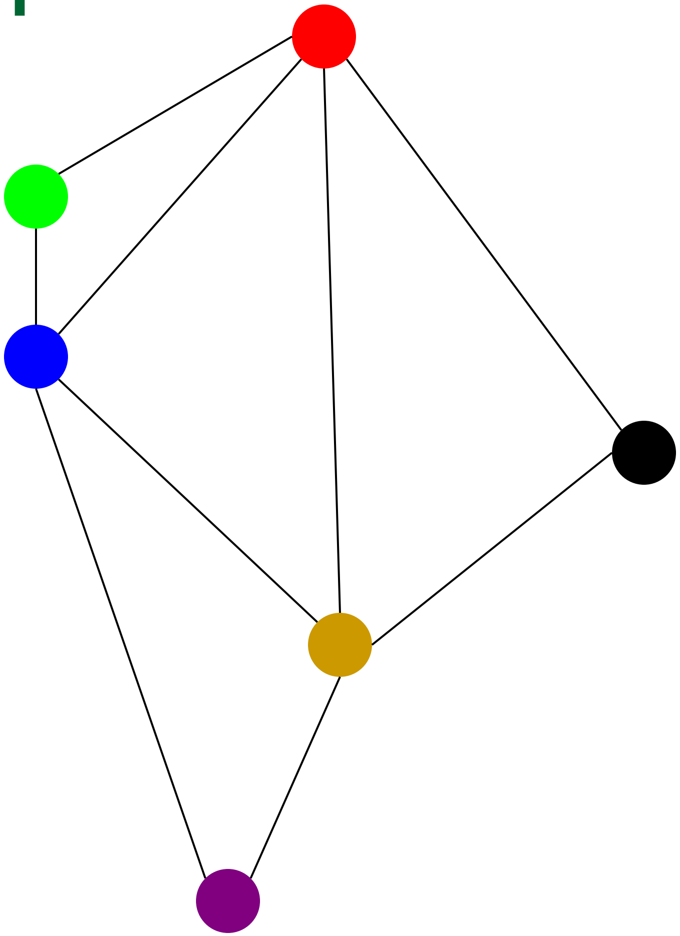
Interval Graph: Branched Genes



Interval Graph: Comparison



Linear genome



Branched genome

Methodology for DNA Sequencing

- The chain termination method (Sanger et al., 1977)
- The chemical degradation method (Maxam and Gilbert, 1977)

DNA Sequencing: History

Sanger method (1977): labeled ddNTPs terminate DNA copying at random points.

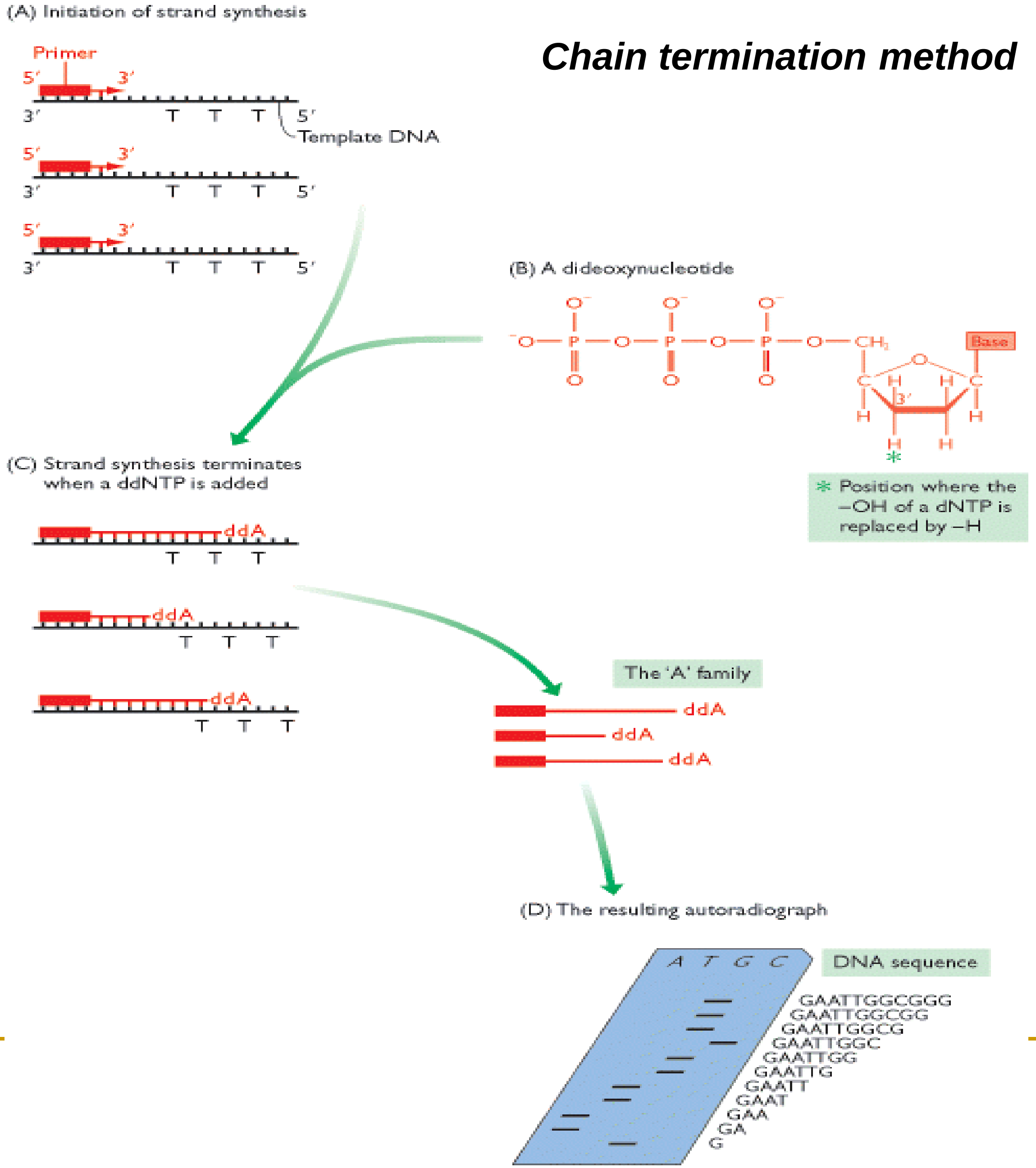
Gilbert method (1977): chemical method to cleave DNA at specific points (G, G+A, T+C, C).

Both methods generate labeled fragments of varying lengths that are further electrophoresed.

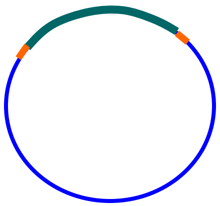
Sanger's Chain Termination Method

- Sanger realized that when a cell copies a DNA, the solution must have supply of all the bases A, T, C and G. If the supply of a certain base is depleted or "starved", the copying process will stop at these base locations. For example, for a sequence ACGTAAGCTA, starving T will produce a mixture of ACG, ACGTAAGC. If the starvation experiment is repeated for all bases, we can obtain what is called a "*Sanger ladder*" – A,AC,ACG,ACGT,ACGTA,ACGTAA, ACGTAAG, ACGTAAGC, ACGTAAGCT and ACGTAAGCTA

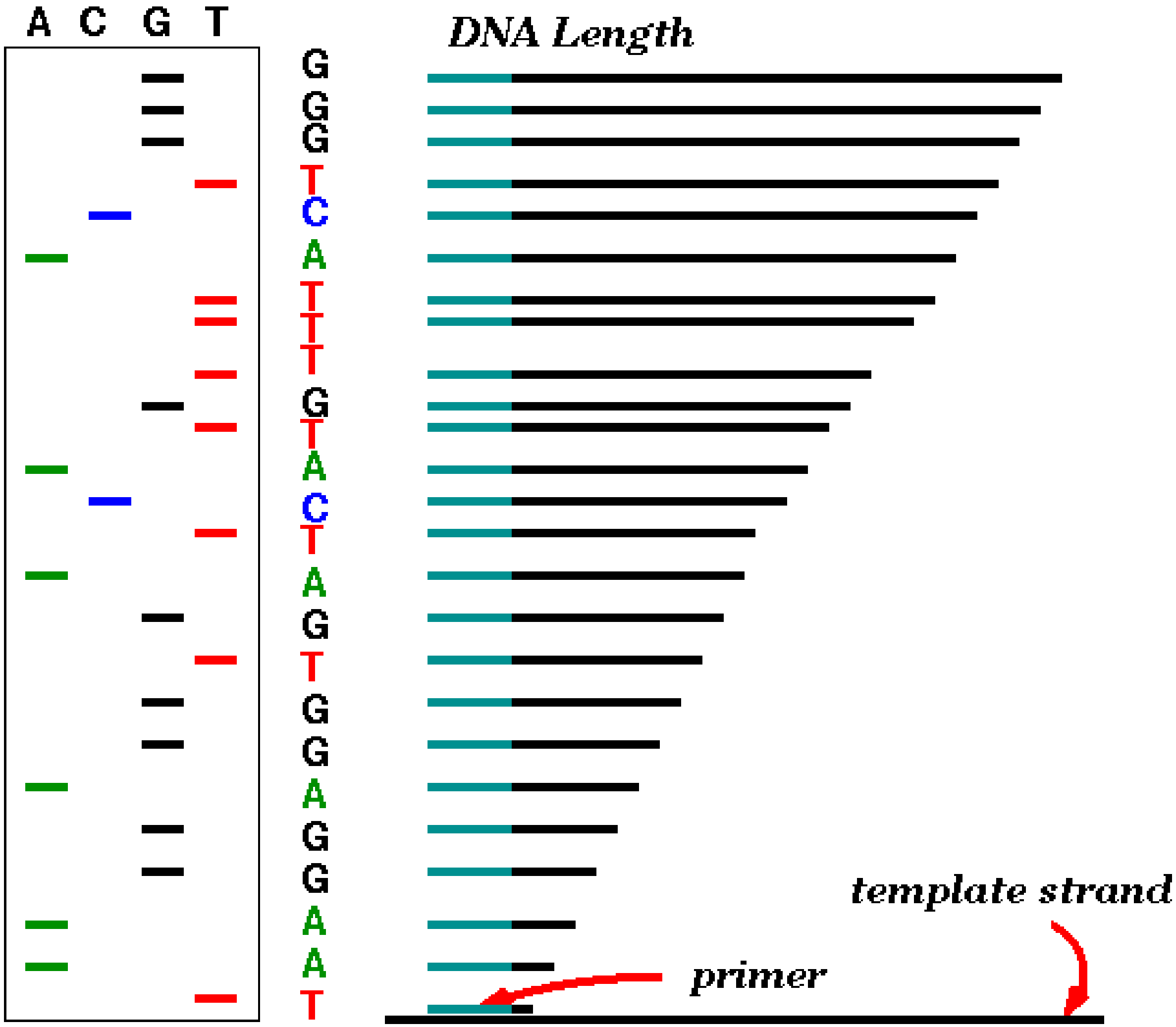
Chain termination method



Sanger Method: Generating Read



1. Start at primer (restriction site)
2. Grow DNA chain
3. Include ddNTPs
4. Stops reaction at all possible points
5. Separate products by length, using gel electrophoresis



Fragment Assembly

- **Computational Challenge:** assemble individual short fragments (reads) into a single genomic sequence (“superstring”)

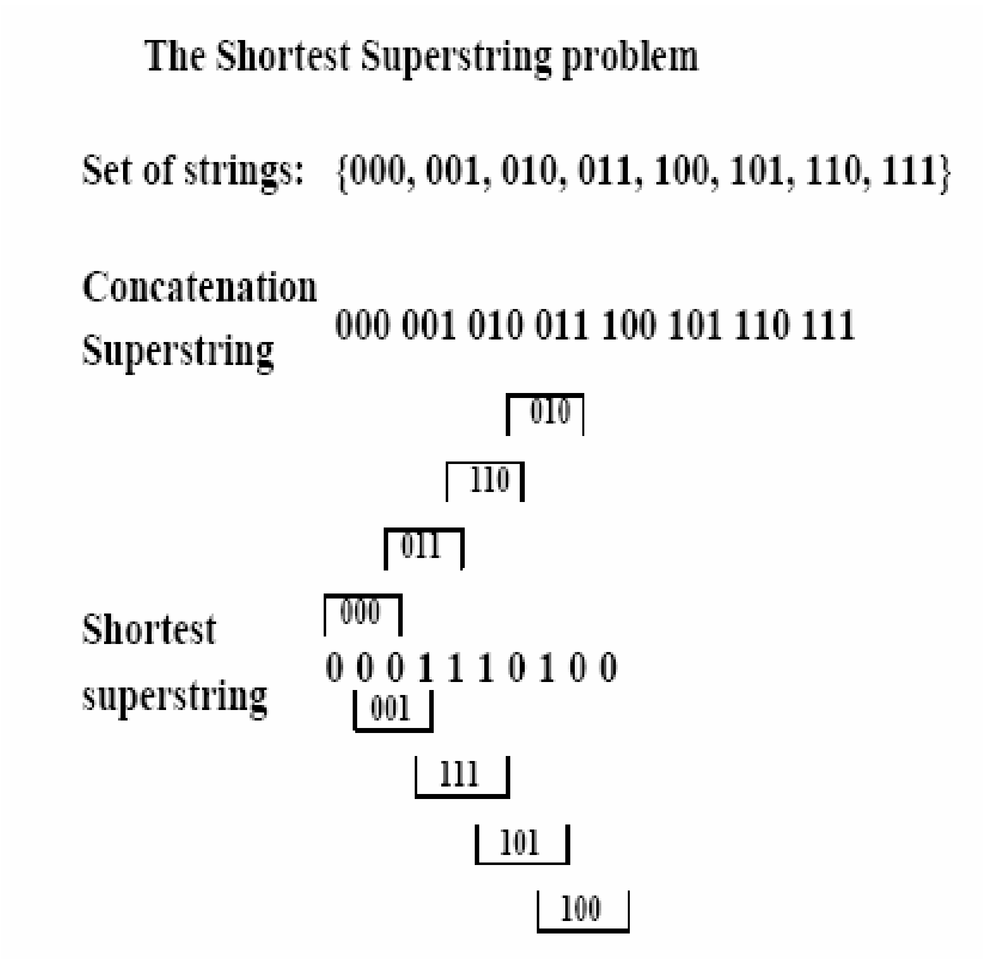
- Until late 1990s the shotgun fragment assembly of human genome was viewed as intractable problem

Shortest Superstring Problem

- Problem: Given a set of strings, find a shortest string that contains all of them
- Input: Strings $s_1, s_2, \ldots., s_n$
- Output: A string s that contains all strings $s_1, s_2, \ldots., s_n$ as substrings, such that the length of s is minimized

- **Complexity:** NP – complete
- **Note:** this formulation does not take into account sequencing errors

Shortest Superstring Problem: Example



The Shortest Superstring problem

Set of strings: {000, 001, 010, 011, 100, 101, 110, 111}

Concatenation Superstring 000 001 010 011 100 101 110 111

Shortest superstring 0 0 0 1 1 1 0 1 0 0

010
110
011
000
001
111
101
100

Reducing SSP to TSP

- Define *overlap* (s_i, s_j) as the length of the longest prefix of s_j that matches a suffix of s_i.

aaaggcatcaaatctaaaggcatcaaa

aaaggcatcaaatctaaaggcatcaaa

What is overlap (s_i, s_j) for these strings?

Reducing SSP to TSP

- Define *overlap* (s_i, s_j) as the length of the longest prefix of s_j that matches a suffix of s_i.

aaaggcatcaaatctaaaggcatcaaa

aaaggcatcaaatctaaaggcatcaaa

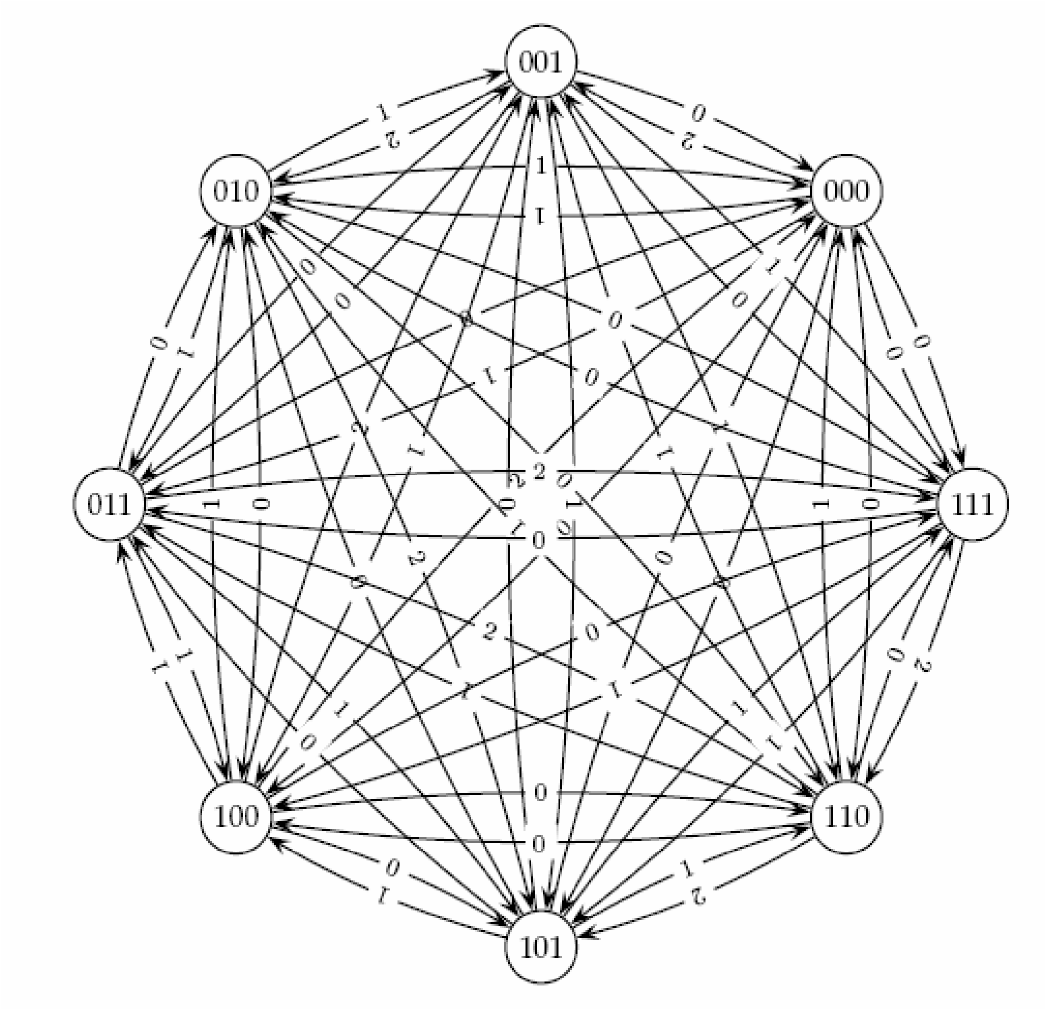
aaaggcatcaaatctaaaggcatcaaa

overlap=12

Reducing SSP to TSP

- Construct a graph with *n* vertices representing the *n* strings s_1, s_2,…., s_n.
- Insert edges of length *-overlap* (s_i, s_j) between vertices s_i and s_j.
- Find the shortest path which visits every vertex exactly once. This is the **Traveling Salesman Problem** (TSP), which is also NP – complete.

Reducing SSP to TSP (cont'd)



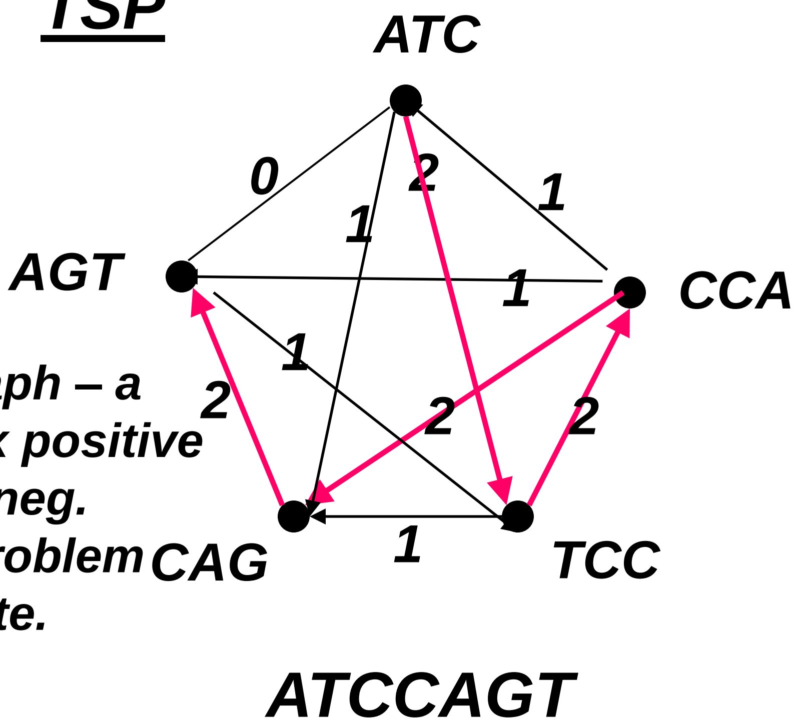
SSP to TSP: An Example

S = { ATC, CCA, CAG, TCC, AGT }

SSP

```
        AGT
      CCA
    ATC
    ATCCAGT
     TCC
       CAG
```

TSP



Complete Graph – a path with max positive score or min neg. score. This problem Is NP-complete.

Greedy Algorithm

Repeatedly merge a pair of strings with maximum overlap until just one string remains .

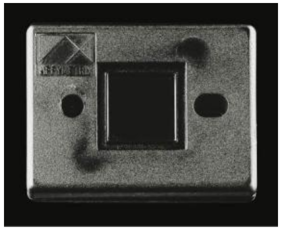
Conjecture: The greedy algorithm at worst generates a string of Twice the length of the optimum string.

Sequencing by Hybridization (SBH): History

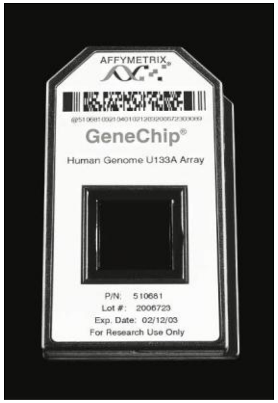
- **1988:** SBH suggested as an alternative sequencing method. Nobody believed it will ever work
- **1991:** Light directed polymer synthesis developed by Steve Fodor and colleagues.
- **1994:** Affymetrix develops first 64-kb DNA microarray

First microarray prototype ***(1989)***

First commercial DNA microarray prototype w/16,000 features ***(1994)***



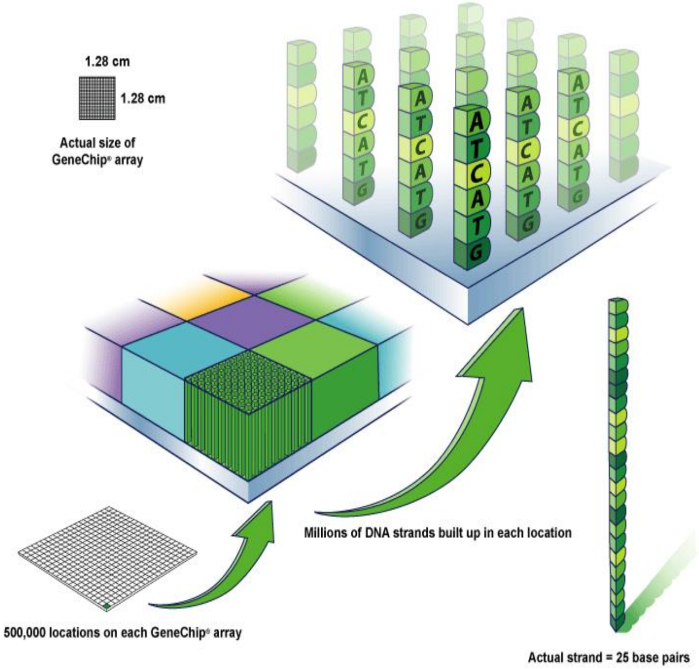
500,000 features per chip ***(2002)***

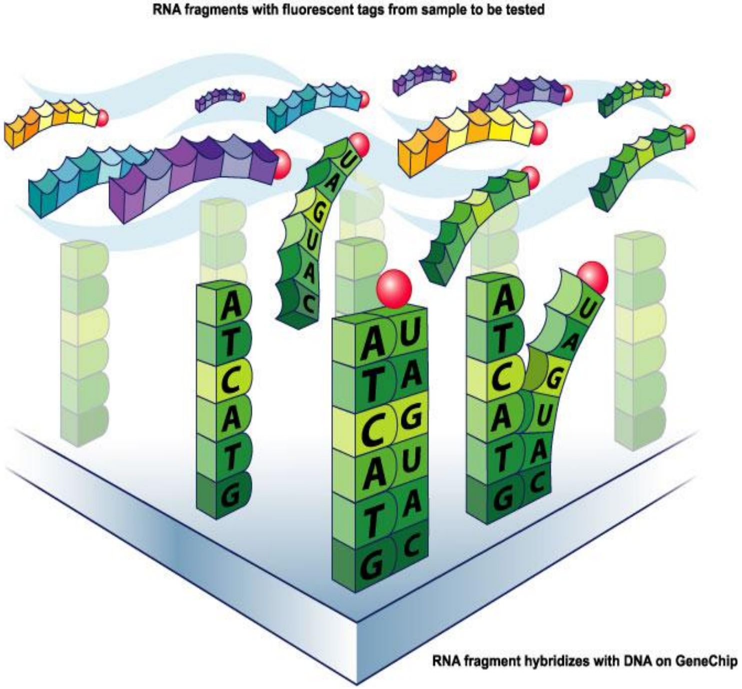


How SBH Works

- Attach all possible DNA probes of length *l* to a flat surface, each probe at a distinct and known location. This set of probes is called the DNA array.
- Apply a solution containing fluorescently labeled DNA fragment to the array.
- The DNA fragment hybridizes with those probes that are complementary to substrings of length *l* of the fragment.

DNA Microarray


1.28 cm
1.28 cm
Actual size of GeneChip® array
ATCATG
Millions of DNA strands built up in each location
500,000 locations on each GeneChip® array
Actual strand = 25 base pairs


RNA fragments with fluorescent tags from sample to be tested
ATCATG
UAGUAC
RNA fragment hybridizes with DNA on GeneChip

Millions of DNA strands build up on each location.

Tagged probes become hybridized to the DNA chip's microarray.

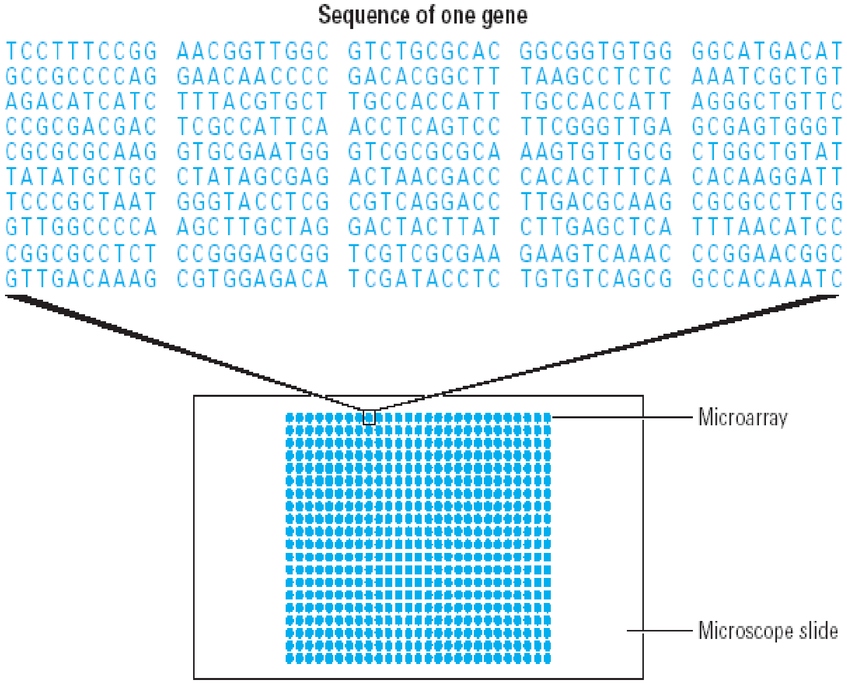
http://www.affymetrix.com/corporate/media/image_library/image_library_1.affx

An Introduction to Bioinformatics Algorithms *www.bioalgorithms.info*

DNA Microarray

Affymetrix

Microarray is a tool for analyzing gene expression that consists of a glass slide.



```
TCCTTTCCGG AACGGTTGGC GTCTGCGCAC GGCGGTGTGG GGCATGACAT
GCCGCCCCAG GAACAACCCC GACACGGCTT TAAGCCTCTC AAATCGCTGT
AGACATCATC TTTACGTGCT TGCCACCATT TGCCACCATT AGGGCTGTTC
CCGCGACGAC TCGCCATTCA ACCTCAGTCC TTCGGGTTGA GCGAGTGGGT
CGCGCGCAAG GTGCGAATGG GTCGCGCGCA AAGTGTTGCG CTGGCTGTAT
TATATGCTGC CTATAGCGAG ACTAACGACC CACACTTTCA CACAAGGATT
TCCCGCTAAT GGGTACCTCG CGTCAGGACC TTGACGCAAG CGCGCCTTCG
GTTGGCCCCA AGCTTGCTAG GACTACTTAT CTTGAGCTCA TTTAACATCC
CGGCGCCTCT CCGGGAGCGG TCGTCGCGAA GAAGTCAAAC CCGGAACGGC
GTTGACAAAG CGTGGAGACA TCGATACCTC TGTGTCAGCG GCCACAAATC
```

Each blue spot indicates the location of a PCR product. On a real microarray, each spot is about 100um in diameter.

How SBH Works (cont’d)

- Using a spectroscopic detector, determine which probes hybridize to the DNA fragment to obtain the *l*–mer composition of the target DNA fragment.

- Apply the combinatorial algorithm (below) to reconstruct the sequence of the target DNA fragment from the *l* – mer composition.

Hybridization on DNA Array

Universal DNA Array

	AA	AT	AG	AC	TA	TT	TG	TC	GA	GT	GG	GC	CA	CT	CG	CC
AA																
AT			ATAG													
AG																
AC												ACGC				
TA											TAGG					
TT																
TG																
TC																
GA																
GT																
GG													GGCA			
GC	GCAA															
CA	CAAA															
CT																
CG																
CC																

DNA target TATCCGTTT (complement of ATAGGCAAA)
hybridizes to the array of all 4-mers:

```
ATAGGCAAA
ATAG
 TAGG
  AGGC
   GGCA
    GCAA
     CAAA
```

l-mer composition

- ***Spectrum (s, l)*** - *unordered* multiset of all possible *(n – l + 1)* *l*-mers in a string *s* of length *n*
- The order of individual elements in *Spectrum (s, l)* does not matter
- For *s* = TATGGTGC all of the following are equivalent representations of *Spectrum (s, 3):*

  {TAT, ATG, TGG, GGT, GTG, TGC}

  {ATG, GGT, GTG, TAT, TGC, TGG}

  {TGG, TGC, TAT, GTG, GGT, ATG}

l-mer composition

- ***Spectrum (s, l)*** - *unordered* multiset of all possible *(n – l + 1)* *l*-mers in a string *s* of length *n*
- The order of individual elements in *Spectrum (s, l)* does not matter
- For *s* = TATGGTGC all of the following are equivalent representations of *Spectrum (s, 3):*

  {TAT, ATG, TGG, GGT, GTG, TGC}

  {ATG, GGT, GTG, TAT, TGC, TGG}

  {TGG, TGC, TAT, GTG, GGT, ATG}
- We usually choose the lexicographically smallest representation as the canonical one.

Different sequences – the same spectrum

- Different sequences may have the same spectrum:

  Spectrum(GTATCT,2)=

  Spectrum(GTCTAT,2)=

  {AT, CT, GT, TA, TC}

The SBH Problem

- Goal: Reconstruct a string from its *l*-mer composition

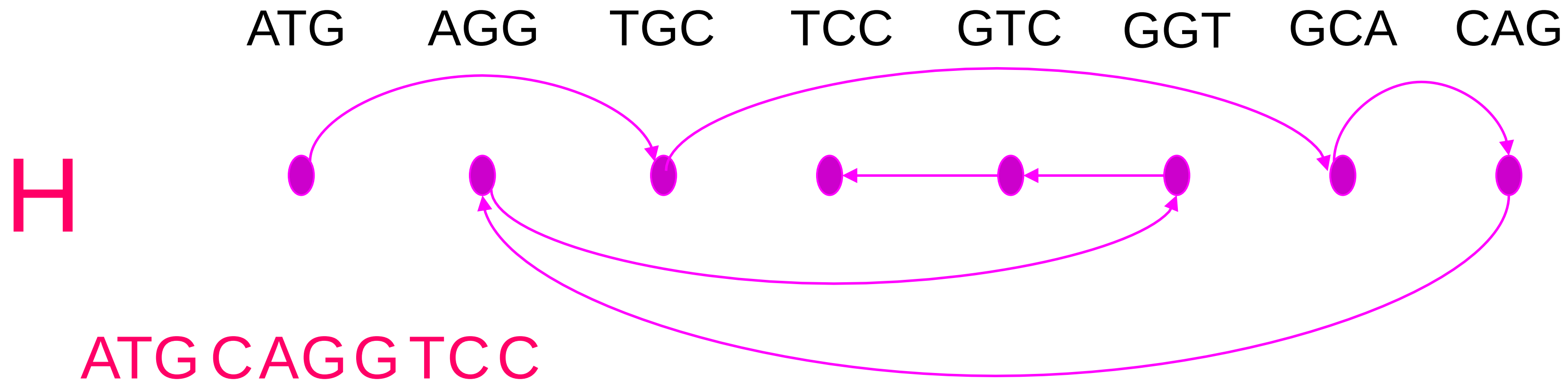
- Input: A set *S*, representing all *l*-mers from an (unknown) string *s*

- Output: String *s* such that *Spectrum (s,l)* = *S*

SBH: Hamiltonian Path Approach

H **is a directed graph with one vertex for each *l*-mer. A vertex *v* is connected to a vertex *w* by a directed edge if and only if the (*l*-1)-length prefix of *v* overlaps with the (*l*-1)-length suffix of *w*.**

S = { ATG AGG TGC TCC GTC GGT GCA CAG }

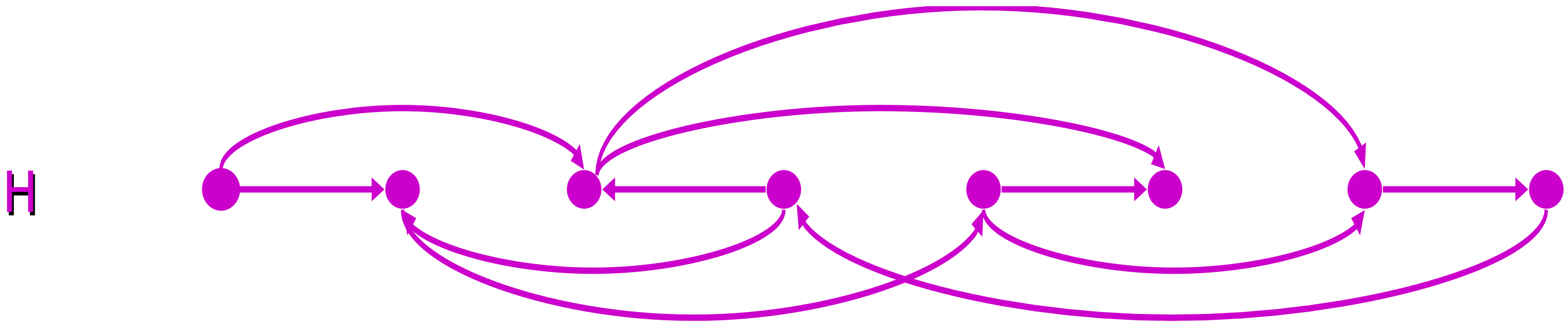


Is the path visited every VERTEX once. Is the path unique for all sequences?

SBH: Hamiltonian Path Approach

A more complicated graph:

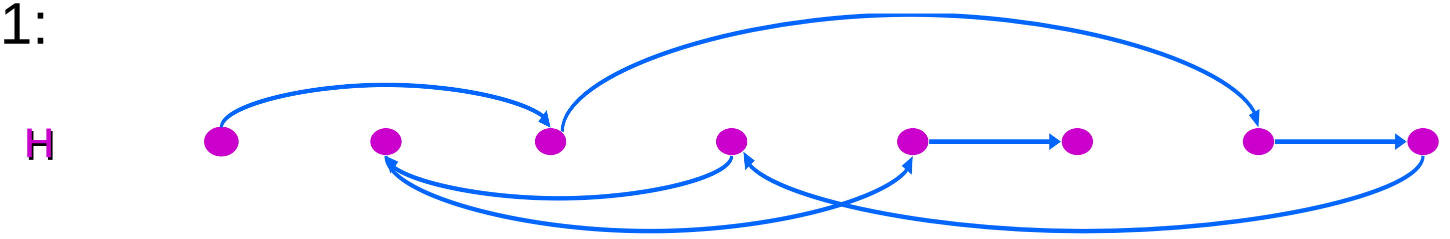
S = { ATG TGG TGC GTG GGC GCA GCG CGT }



SBH: Hamiltonian Path Approach

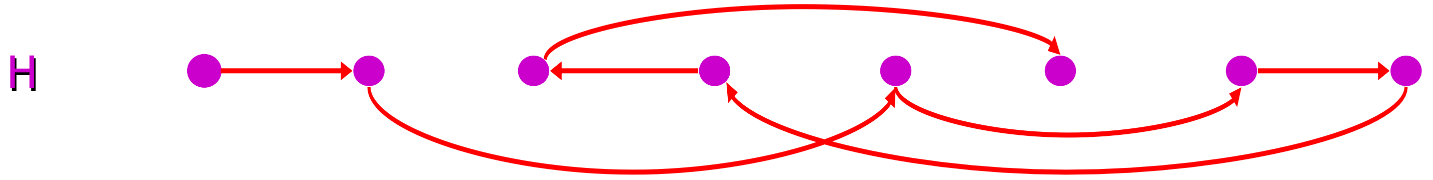
S = { ATG TGG TGC GTG GGC GCA GCG CGT }

Path 1:



ATGCGTGGCA

Path 2:



ATGGCGTGCA

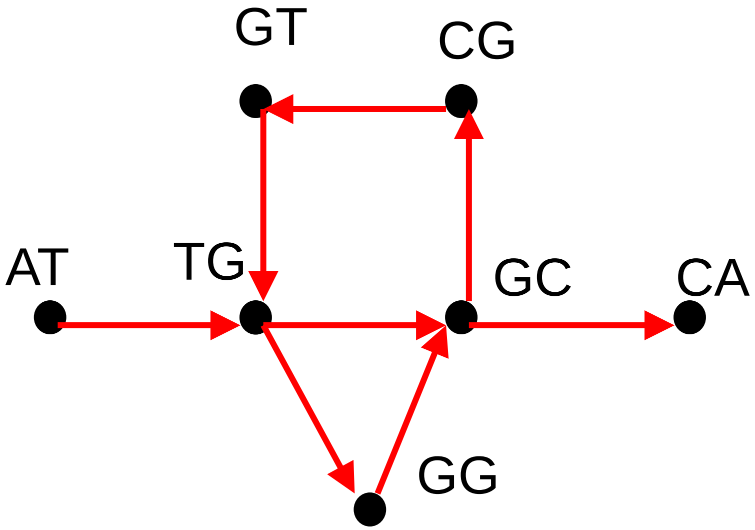
But the Hamiltonian path problem is NP-complete

SBH: Eulerian Path Approach leads to linear time algorithm

S = { ATG, TGC, GTG, GGC, GCA, GCG, CGT }

Vertices correspond to (*l* – 1) – mers : { AT, TG, GC, GG, GT, CA, CG }

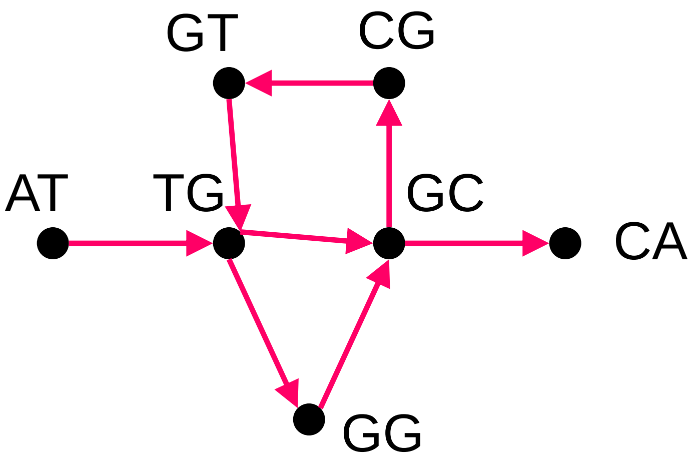
Two vertices ***v*** and ***w*** are connected by a directed edge if the spectrum contains a ***l***-mer with ***v*** as a *l*-1 prefix and w as a *l*-1 suffix of the l-mer, respectively. Thus, the edges correspond to *l* – mers from *S.*



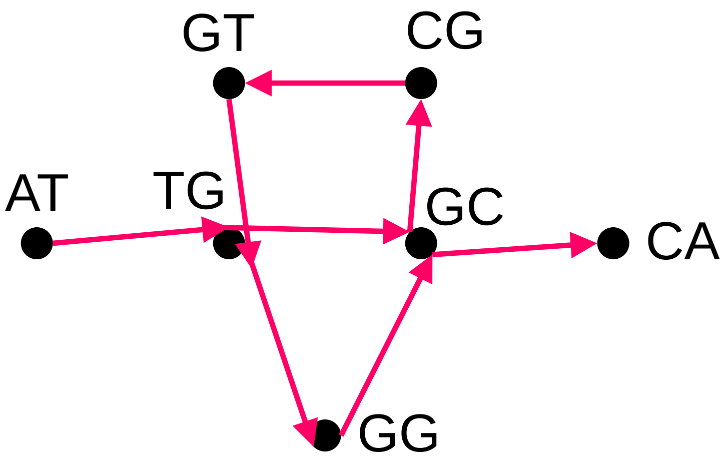
A DNA fragment containing all the *l*-mers from *S* corresponds to a path visited by every EDGE once

SBH: Eulerian Path Approach

S = { AT, TG, GC, GG, GT, CA, CG } corresponds to two different paths. Note a vertex could be visited more than once.



ATGGCGTGCA



ATGCGTGGCA

Euler Theorem

- A graph is balanced if for every vertex the number of incoming edges equals to the number of outgoing edges:

$$in(v)=out(v)$$

- **Theorem**: *A connected graph is Eulerian if and only if each of its vertices is balanced.*

Euler Theorem: Proof

- Eulerian → balanced

for every edge entering *v* (incoming edge) there exists an edge leaving *v* (outgoing edge). Therefore

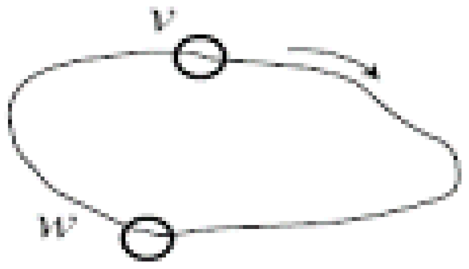
$$in(v)=out(v)$$

- Balanced → Eulerian

???

Algorithm for Constructing an Eulerian Cycle

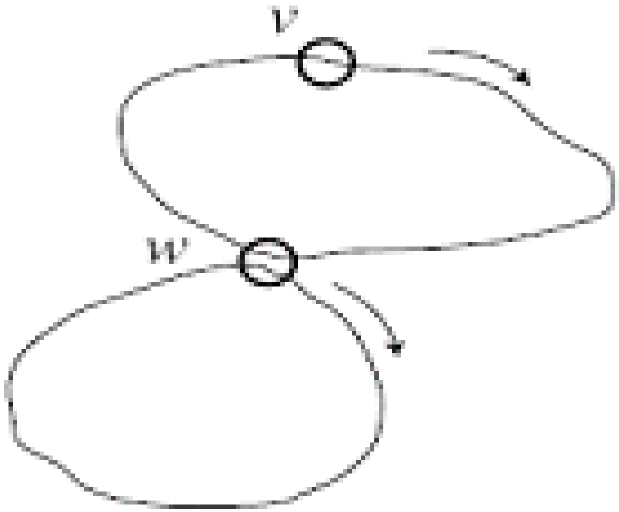
a. Start with an arbitrary vertex *v* and form an arbitrary cycle with unused edges until a dead end is reached. Since the graph is Eulerian this dead end is necessarily the starting point, i.e., vertex *v*.



(a)

Algorithm for Constructing an Eulerian Cycle (cont'd)

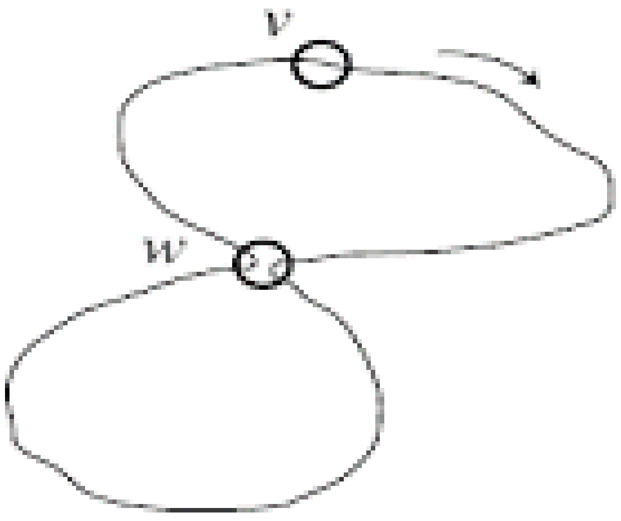
b. If cycle from (a) above is not an Eulerian cycle, it must contain a vertex *w*, which has untraversed edges. Perform step (a) again, using vertex *w* as the starting point. Once again, we will end up in the starting vertex *w.*



(b)

Algorithm for Constructing an Eulerian Cycle (cont'd)

c. Combine the cycles from (a) and (b) into a single cycle and iterate step (b).



(c)

Euler Theorem: Extension

- **Theorem**: *A connected graph has an Eulerian path if and only if it contains at most two semi-balanced vertices and all other vertices are balanced.*

Some Difficulties with SBH

- **Fidelity of Hybridization:** difficult to detect differences between probes hybridized with perfect matches and 1 or 2 mismatches
- **Array Size:** Effect of low fidelity can be decreased with longer *l*-mers, but array size increases exponentially in *l.* Array size is limited with current technology.
- **Practicality:** SBH is still impractical. As DNA microarray technology improves, SBH may become practical in the future
- **Practicality again**: Although SBH is still impractical, it spearheaded expression analysis and SNP analysis techniques

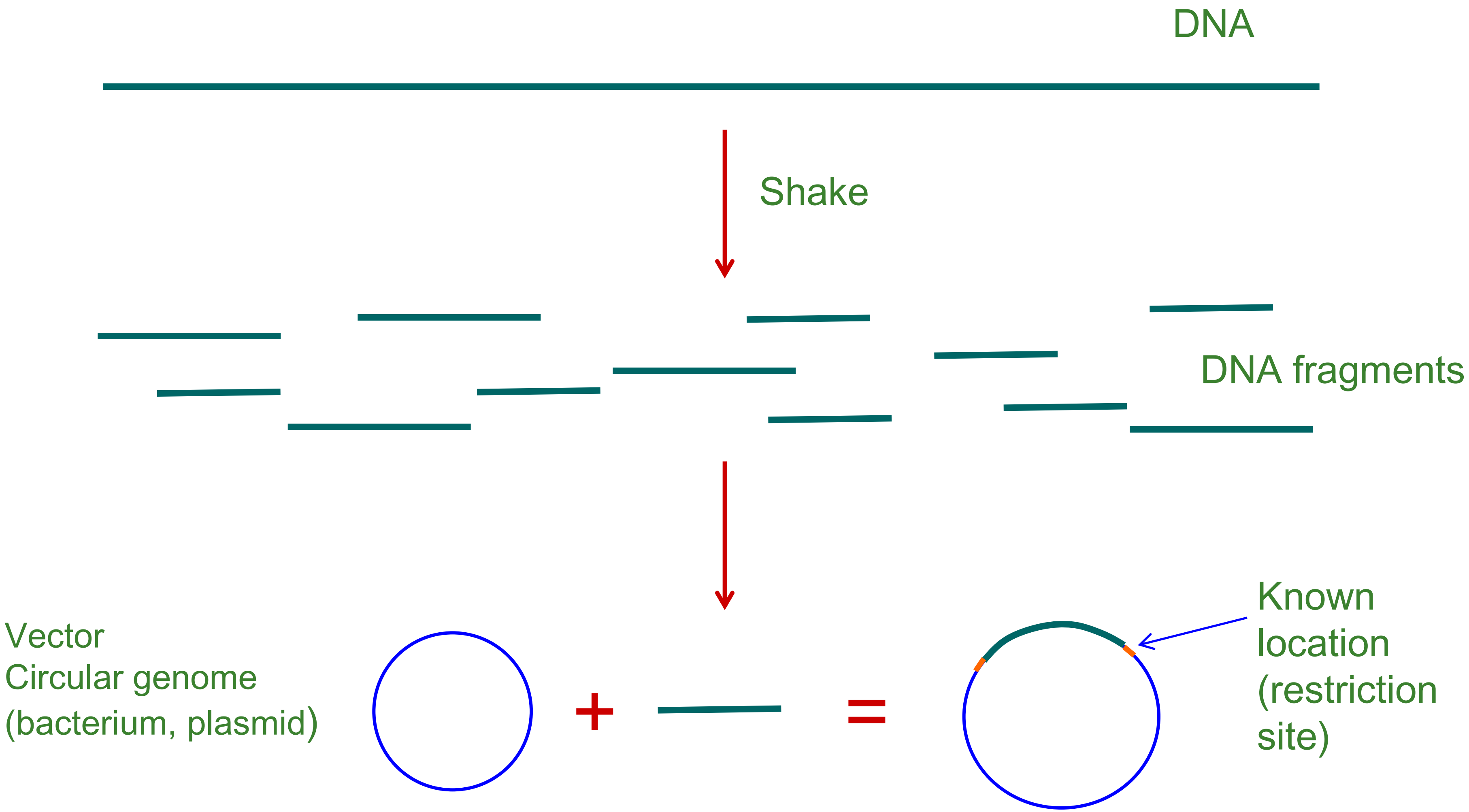
What can we do?

- **Two approaches:**
  - Approximation Algorithms
  - Evolutionary Algorithms

Shotgun Sequencing

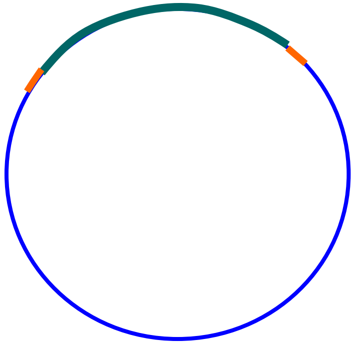
- This method allowed Sanger to sequence a 5386 –nucleotide virus in 1977 and a Nobel prize in the same year.
- Modern *Shotgun Sequencing* starts with a large sample of DNA. Samples of size less than 500 nuleotides are removed (a process called “sonicated”). The remaining samples are called *inserts* which are multiplied billion times (using cloning) so that the ladders produced by Sanger’s method can be read.

Traditional DNA Sequencing

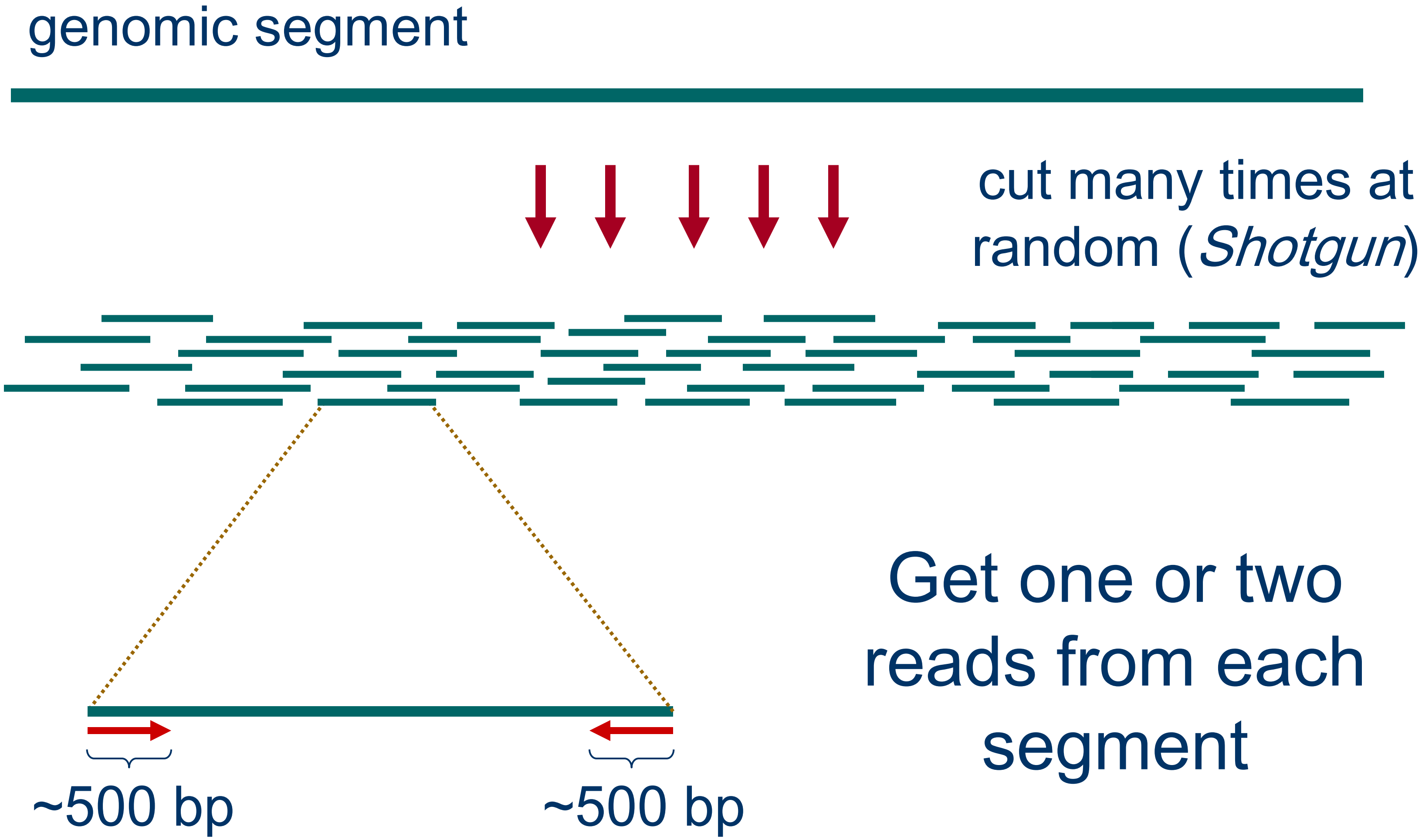


Different Types of Vectors

VECTOR	**Size of insert (bp)**
Plasmid	2,000 - 10,000
Cosmid	40,000
BAC (Bacterial Artificial Chromosome)	70,000 - 300,000
YAC (Yeast Artificial Chromosome)	> 300,000 Not used much recently

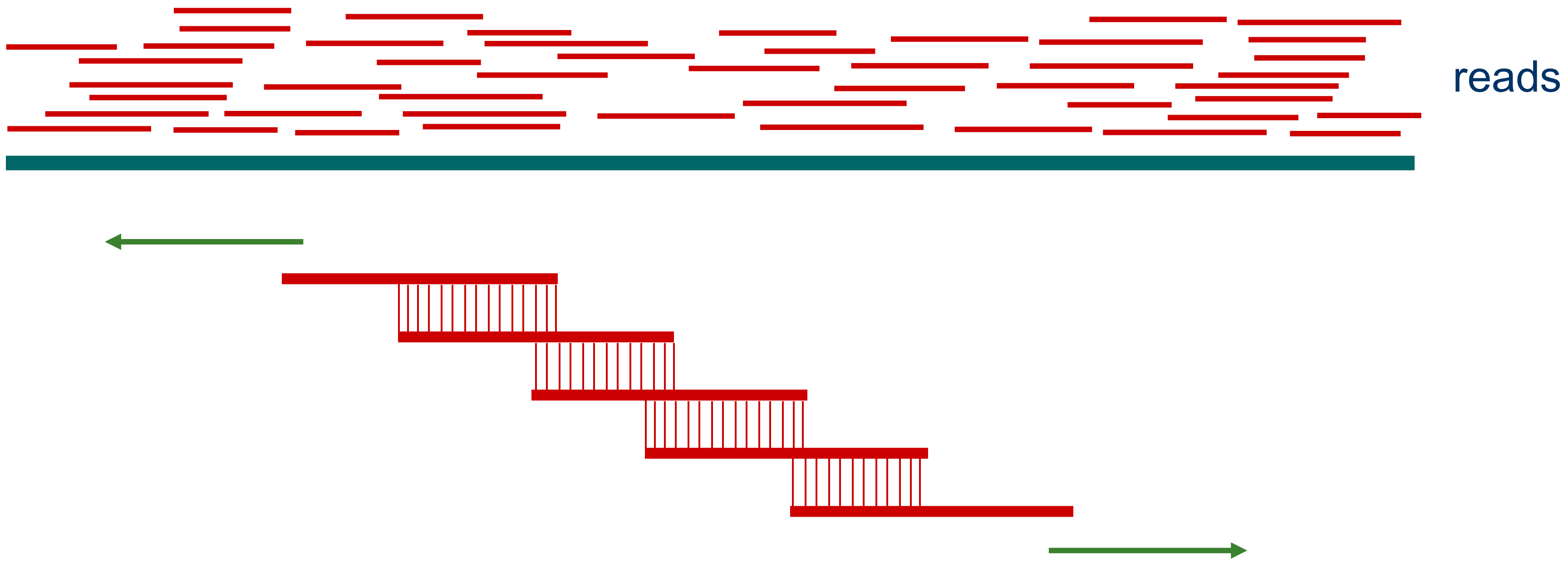


Shotgun Sequencing



Get one or two reads from each segment

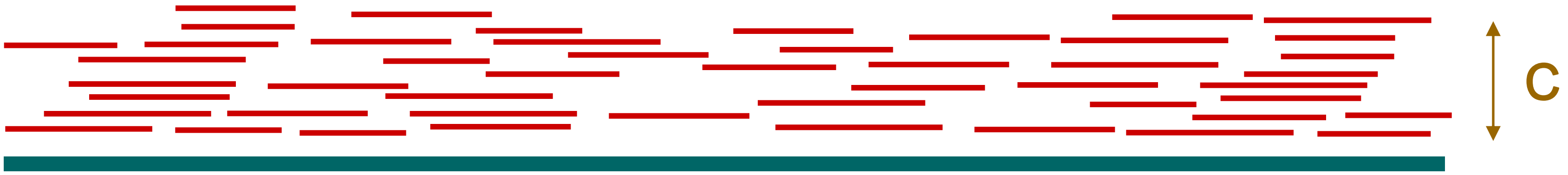
Fragment Assembly



Cover region with ~7-fold redundancy

Overlap reads and extend to reconstruct the original genomic region

Read Coverage



Length of genomic segment: ***L***

Number of reads: ***n***

Length of each read: ***l***

Coverage $C = n\, l \,/\, L$

How much coverage is enough?

Lander-Waterman model:

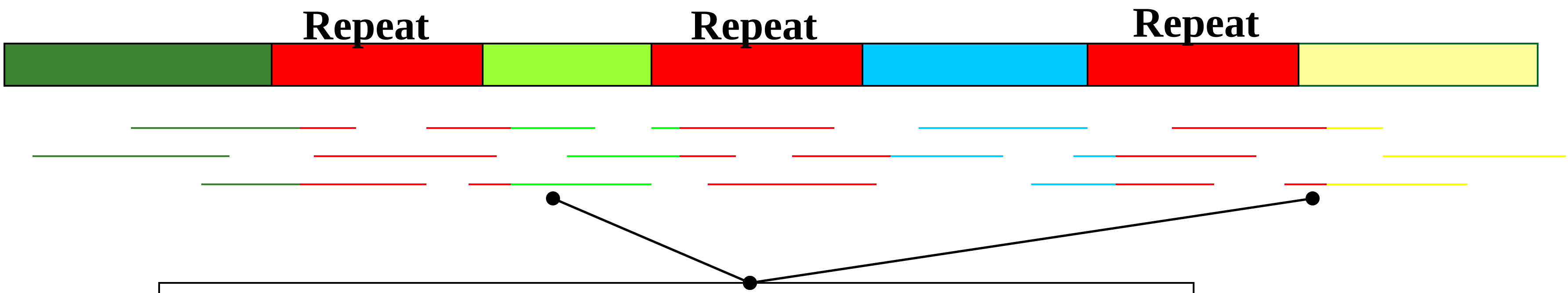
Assuming uniform distribution of reads, *C*=10 results in 1 gapped region per 1,000,000 nucleotides

Challenges in Fragment Assembly

- Repeats: A **major** problem for fragment assembly
- > 50% of human genome are repeats:

  - over 1 million *Alu* repeats (about 300 bp)

  - about 200,000 LINE repeats (1000 bp and longer)



Green and blue fragments are interchangeable when assembling repetitive DNA

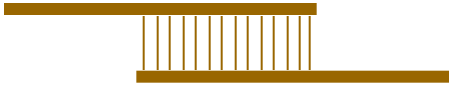
Repeat Types

- Low-Complexity DNA (e.g. ATATATATACATA…)
- Microsatellite repeats $(a_1 \ldots a_k)^N$ where k ~ 3-6
  (e.g. CAGCAGTAGCAGCACCAG)
- Transposons/retrotransposons
  - SINE Short Interspersed Nuclear Elements
    (e.g., *Alu*: ~300 bp long, 10^6 copies)
  - LINE Long Interspersed Nuclear Elements
    ~500 - 5,000 bp long, 200,000 copies
  - LTR retroposons Long Terminal Repeats (~700 bp) at each end
- Gene Families genes duplicate & then diverge
- Segmental duplications ~very long, very similar copies

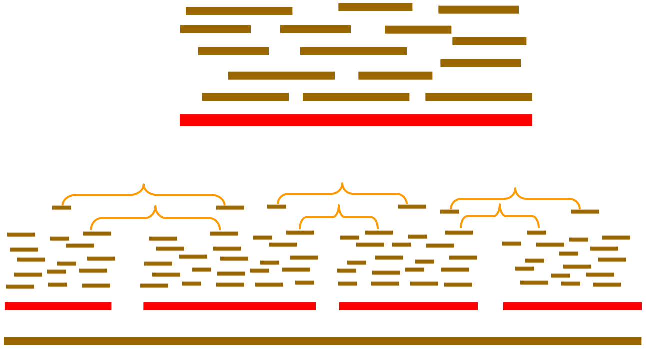
Overlap-Layout-Consensus

Assemblers: ARACHNE, PHRAP, CAP, TIGR, CELERA

Overlap: find potentially overlapping reads



Layout: merge reads into contigs and contigs into supercontigs



Consensus: derive the DNA sequence and correct read errors

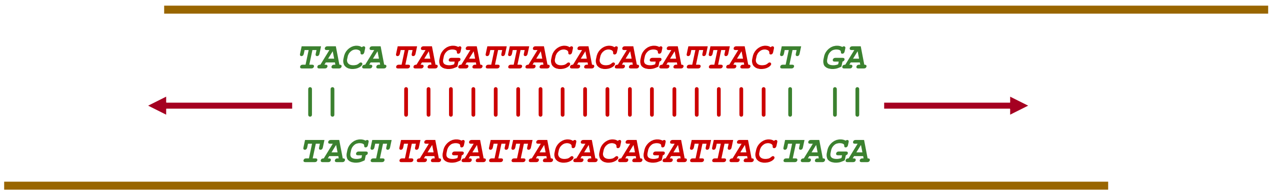
..ACGATTACAATAGGTT..

Overlap

- Find the best match between the suffix of one read and the prefix of another
- Due to sequencing errors, need to use dynamic programming to find the optimal *overlap alignment*
- Apply a filtration method to filter out pairs of fragments that do not share a significantly long common substring

Overlapping Reads

- ***Sort all k-mers in reads (k ~ 24)***
- ***Find pairs of reads sharing a k-mer***
- ***Extend to full alignment – throw away if not >95% similar***



TACA TAGATTACACAGATTACT GA

TAGT TAGATTACACAGATTAC TAGA

Overlapping Reads and Repeats

- A *k*-mer that appears N times, initiates N^2 comparisons

- For an *Alu* that appears 10^6 times → 10^{12} comparisons – too much

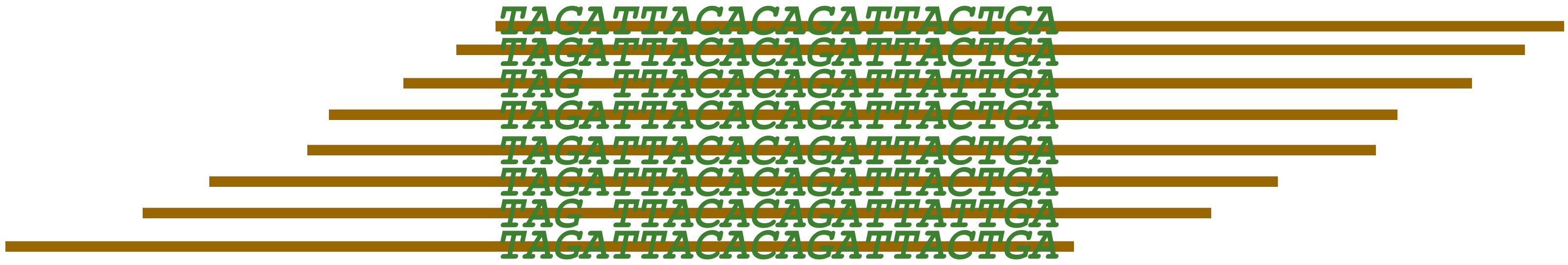
- **Solution:**

  Discard all *k*-mers that appear more than

  $t \times$ Coverage, ($t \sim 10$)

Finding Overlapping Reads

Create local multiple alignments from the overlapping reads



```
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAGATTACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACAGATTACTGA
```

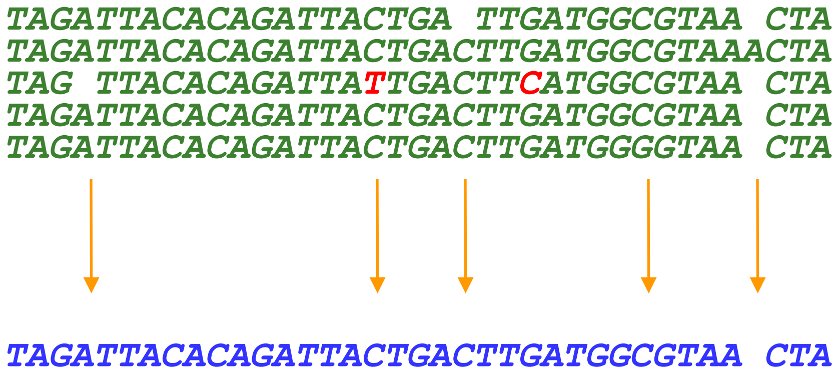
Layout

- Repeats are a major challenge
- Do two aligned fragments really overlap, or are they from two copies of a repeat?
- Solution: repeat masking – hide the repeats!!!
- Masking results in high rate of misassembly (up to 20%)
- Misassembly means alot more work at the finishing step

Consensus

- A consensus sequence is derived from a profile of the assembled fragments
- A sufficient number of reads is required to ensure a statistically significant consensus
- Reading errors are corrected

Derive Consensus Sequence



```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGGGTAA CTA

TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

- Derive multiple alignment from pairwise read alignments

- *Derive each consensus base by weighted voting*

EULER - A New Approach to Fragment Assembly

- Traditional “overlap-layout-consensus” technique has a high rate of mis-assembly
- EULER uses the Eulerian Path approach borrowed from the SBH problem
- Fragment assembly without repeat masking can be done in linear time with greater accuracy

Conclusions

- Graph theory is a vital tool for solving biological problems

- Wide range of applications, including sequencing, motif finding, protein networks, and many more

References

- Simons, Robert W. *Advanced Molecular Genetics Course*, UCLA (2002). http://www.mimg.ucla.edu/bobs/C159/Presentations/Benzer.pdf

- Batzoglou, S. *Computational Genomics Course*, Stanford University (2004). http://www.stanford.edu/class/cs262/handouts.html